03 | 程序员的测试与测试人员的测试有什么不同？

2021-08-09 郑晔

《程序员的测试课》 课程介绍 >

讲述：郑晔

时长 11:36 大小 10.63M

你好，我是郑晔！

前面用了两讲的篇幅，我们一起一步一步地用带测试的方式完成了一个项目，现在相信你已经对如何在实际工作中编写测试有了一个初步的认识。有了实践的根基，我们还需要对如何编写测试有一个更全面地理解，以便日后能够更好地应对各种场景。

关于测试，许多程序员的第一个问题就是：测试不是测试人员的工作吗？如果我把测试写了，那是不是就抢了测试人员的工作呢？

不瞒你说，之所以我要把这个话题放在专栏前面讲，一个重要的原因就是我当年真的就这么想过。好，今天我们就来聊聊程序员的测试和测试人员的测试究竟有哪些不一样的地方。

程序员的测试能否替代测试人员的测试？

我给你讲一个我在职业生涯初期的故事。那时候，我刚刚踏上自己的程序员精进之路，我不断地寻找着各种能够更好地写程序的方式。当我意识到测试对于编程的重要性时，我就开始有意识地在写代码的时候编写测试，尽我所能把各种场景都考虑到。作为一个骄傲的程序员，我总是希望自己的代码是无懈可击的。

有一次，我写了一个协议的解析器，我把各种字段缺失或不正确的场景都处理了。结果交给测试同学后，他上来就发了一个空包，然后我的代码就崩溃了。我当时的第一反应是，你怎么能这么做？测试同学却反问，我为什么不能这么做？

是啊，为什么不能呢？测试同学只要能做到，他就可以这么做。而且，只要测试同学能做到，其他人也可能做到。

可以说，这件事彻底改变了我对测试人员的认识。相信很多人和从前的我一样有个偏见，认为测试同学不过是做一些简单的验证，或者只是因为自己时间不充足，有些细节没考虑到，让他们给抓住点小问题。作为程序员，只要自己认真了，其实就没测试什么事了。然而，这次的事告诉我，即便我全力以赴了，测试同学依然可以发现问题。

从此，我对测试人员的看法彻底转变了。在我随后的职业生涯中我发现，只要团队里有合格的测试人员，他总会以你想不到的角度，发现系统中意想不到的问题——即便团队已经写了很多的测试。

说到这，你就可以放心了，即便我们程序员把测试都写好了，测试人员也不会失业，他们总能找到问题。但下一个问题就随之而来了，测试人员的测试和程序员的测试到底有什么不一样，以至于即便是程序员已经很努力了，依然很难做到对测试场景全面的覆盖呢？

答案很简单，因为视角不同。

程序员的出发点是实现，而测试人员的出发点是业务。把这话翻译成你更熟悉的测试术语，那就是程序员的关注点是白盒测试，而测试人员则是黑盒测试。

程序员关注到的测试是站在实现的角度，即便我们能够先去设计测试场景，即便我们有测试覆盖率帮我们查缺补漏，但我们所做的一切都是建立在一个共同的前提下：我们要把代码写出来。

而测试人员则不同，他们并不关心代码是怎么实现的，他们只是站在业务的视角在想问题，他们考虑更多的是这个系统可以怎么用。只要二者的出发点不同，对于同样的事物，总会看到不同的东西。不然的话，人类社会哪有那么多的争论。

你或许会想，那我也从业务的角度去想，是不是就能获得测试人员的视角呢？

我要说，**从业务角度思考，确实是我们向测试人员学习的一个重要方向。**但同样不要指望你换个角度思考一下就能把测试人员代替了。人的注意力是有限的，作为一个程序员，我们会把更多的时间放在关于技术实现的思考上，**我们在发现问题上的训练强度是远远不够的。**所以，人们常说，别用你的业余爱好去挑战别人吃饭的本事。

程序员做测试，测试人员也做测试，那是不是测试人员的工作量就小多了？实际上，只要你稍微和测试同学交流一下你就会发现，在实际的工作中，**大部分测试同学根本没有机会使出全力。**

在测试的分类中，有一种测试叫探索性测试，也就是测试人员竭尽所能去对系统去做测试。不过，虽然有这么个分类，但大多数测试人员并没有机会去做这种测试。不是因为他们偷懒，而是由于大部分系统的基础质量不高，造成的结果就是，测试同学的大部分工作找到的都是极其简单的 bug。换言之，如果程序员能够把自己的测试做好，很多问题就应该被消灭在萌芽状态，根本不应该到测试同学这里。

如果程序员能够提交一个经过自己测试的系统，测试同学才有机会让自己从日常琐碎的工作摆脱出来，去竭尽全力地测试一个系统。**不是测试同学不努力，实在是系统太差劲。**

在我工作过的测试做得比较好的团队，软件质量整体上来说，确实要好上很多。测试同学有机会进行各种探索性测试，因为基础的问题都会在程序员的测试中被覆盖了。

好，到这里，我们也就回答了一开头的那个问题：**程序员的测试不能够替代测试人员的测试。**我们也不用替测试人员担心他们的职业前景了。那反过来，既然在测试方面，测试人员还是有着自己的优势，那是不是我们可以从他们身上学到点什么呢？

向测试人员学习

首先，我要帮你纠正一个典型的误区。有一些人认为，测试做得好要依赖于工具。确实，今天的测试已经不像过去那么纯手工了，各种工具层出不穷，甚至很多项目为了测试要开

发自己的测试工具。**但无论是什么样的测试工具，都只是提升效率的一种手段。**如果没有背后的测试思维支撑，再好的工具，也是没用的。

与其纠结于寻找更好的工具，更重要的是要向测试人员学习他们的思维方式。

前面我们已经提到了一点，测试人员拥有业务视角，这是最值得程序员去学习的。通常来说，测试人员对于业务的理解都会很深刻。在我之前的经历里，如果一个团队缺少业务分析师，有时候，我会让一个测试人员顶上去，而且效果往往还不错。

程序员对业务视角的忽略是一个普遍存在的问题，但这也是一个程序员必须要突破的关口。实际上，让程序员拥有业务视角不仅仅是测试的需求，同样也是写好代码的要求。现在流行的 DDD 设计方法的核心就是要让业务人员和开发团队使用一样的通用语言（Ubiquitous Language）。由此我们知道，写代码要尽可能用业务语言写代码。

有了业务视角，再深入一步，我们要学会设计测试。

设计测试的一个关键点是找到更多的测试场景。这里面我们说的测试场景，你可以把它理解成一个大的分支条件。其实很多时候，程序员与测试人员的差距，从测试场景这一步已经开始显现出来了。

很多测试场景程序员压根就没想到，所以，就很难说进一步地去测试了。在前面我自己的经历里，我只想到了协议包不正确的各种情况。但空协议包这种场景在我的思考里压根就没有，所以，我发现不了其中的问题就再正常不过了。

针对测试场景，我们还需要考虑各种情况。在这里我们说的各种情况，你可以把它理解成在一个测试场景下的各种小分支。通常来说，正常情况大家都能想到怎么解决，程序员欠缺的往往是各种异常情况的处理。

举个例子，同样是注册的场景，程序员都会考虑到正常注册的情况。但如果注册的用户名里包含各种符号该怎么办，甚至有些测试人员会想到如果字符串超长的情况该怎么办（比如几 K 字节的字符串），这些都是对不同的异常情况的思考。

诚如前面所说，很难指望每个程序员都能把所有的情况考虑到，但**程序员每多想到一点，软件质量就能多提高一点。**

在《10x 程序员工作法》中我曾经讲过，每个需求都应该有验收条件。验收条件是很多测试人员设计测试的出发点，这也是我们可以向测试人员学习的最直接方式，当然，前提条件是你们的团队有验收条件。如果没有，那你需要赶紧去建立这项团队实践。

其实在实际工作中，程序员与测试人员如果能够工作在一个团队里，那就有一个更简单的做法来提升软件质量，就是测试人员设计好了测试用例之后在团队内部做一个分享，让相关的程序员能够有一个参考去编写自己的测试。

不过，请你放心，虽然他们已经把测试用例分享给你了，在测试过程中，他们还会发现更多新的测试用例。如果程序员有了编写测试的习惯，那测试人员在测试过程中发现的问题，也可以成为一个新的测试项，成为程序员编写测试的一部分，我们可以用代码把这个用例固化下来。

测试人员把用例分享给程序员，程序员用代码固化新的测试用例，这样，测试人员和开发人员之间就形成了一个良好的互动，我们也就有机会让软件的质量越做越好。

总结时刻

今天，我们谈到可能是不写测试的程序员关于测试问得最多的一个问题：程序员写测试，那测试人员怎么办？

程序员和测试人员拥有不同的视角，程序员更关注实现，而测试人员更关注业务，所以，即便程序员编写测试，也很难覆盖所有的情况。实际上，即便是测试人员也不敢说自己能够覆盖所有的情况。

目前大多数团队的情况是，测试人员并没有得到充分的发挥。只有程序员做好了自己的测试，测试人员才能从日常琐碎的验证工作中解脱出来，去做更有价值的测试。

在测试问题上，程序员应该向测试人员学习，与工具相比，更重要的是思维方式。我们可以像测试人员一样从测试场景入手，多考虑各种情况，尤其是异常情况。需求的验收条件是一个很好的测试起始点。

在团队中，测试人员可以把自己的测试用例分享给程序员，而程序员可以把新的测试用例用代码的方式固化下来，二者就此可以形成良好的互动。

其实，具体确定测试用例的方法有很多，比如边界值分析、等价类划分等等，这都是测试同学会更深入了解的内容，如果这一讲的内容让你对测试产生一些兴趣，你不妨去找本书读读，比如《软件测试的艺术》，或者找个测试同学深入请教一番。

如果今天的内容你只能记住一件事，那请记住：**测试从测试场景入手，多考虑各种情况，尤其是异常情况。**

思考题

今天我们讲了程序员和测试人员之间的关系，你在实际工作中，从测试同学身上学到了哪些东西呢？欢迎在留言区分享你的经历。

分享给需要的人，Ta订阅后你可得 20 元现金奖励

赞 1 提建议

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 02 | 实战：实现一个 ToDo 的应用（下）

下一篇 04 | 自动化测试：为什么程序员做测试其实是有优势的？

精选留言 (4)

写留言

2021-08-10

从「10x程序员工作法」到「代码之丑」然后来到了「程序员的测试课」，昨天又情不自禁把「软件设计之美」纳入学习中。 非常喜欢郑老师的风格。案例生动，简单意骇。从10x到代码之丑，明显发现了自己的问题。原来自己虽然很努力学习。但一直在错误的道路上。感谢老师提供了一条清晰明确的道路。对未来从此不再迷惘。

展开

作者回复: 欢迎回归

webmin
2021-08-09

今天的课程帮我修正了对测试工作的认知，我以往认为测试人员的研发能力如果没有研发人员的能力强的话，怎么能发现研发人的问题。

展开

作者回复: 很多问题的出现都是脑子里的开关，拨过去就不一样了

胖虫子
2021-08-09

终于遇到个能理解测试的开发了，有时候提交的东西都是一堆明显问题，此时修改来修改去，真的留给测试人员专心去想，专心去测的时间已不多了

展开

作者回复: 扩大上下文，理解不同角色的价值。

1

webmin
2021-08-09

测试人员通常采用反证法，用无法证伪来达到当下足够正确。
程序员大多时候习惯逻辑推理，只是这个推理过程的严谨性无法和数学的证明过程相比。

作者回复: 我在《软件设计之美》中讲过，一开始人们想通过证明的方式验证程序的正确性，不过，这条路在实践中没有走通，只能靠测试了。